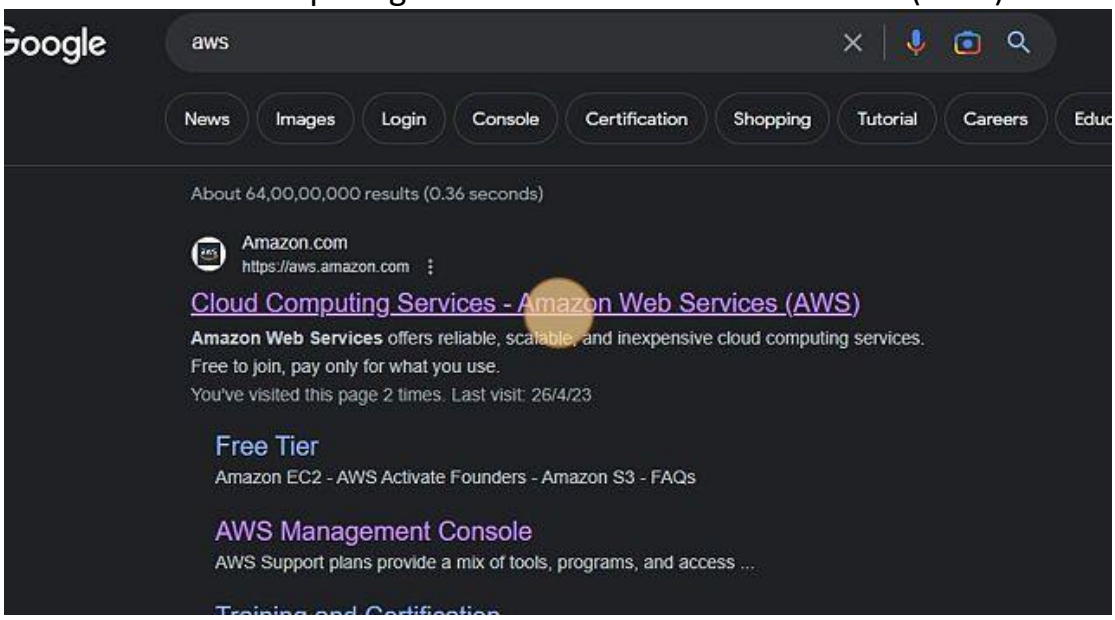
AWS Workflow

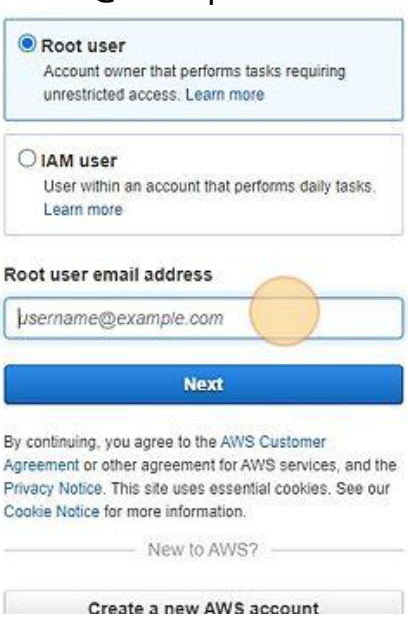
1. Navigate to www.google.com/search?q=aws&oq=&aqs=chr...

2. Click "Cloud Computing Services - Amazon Web Services (AWS)"2



3. Click "Sign In"

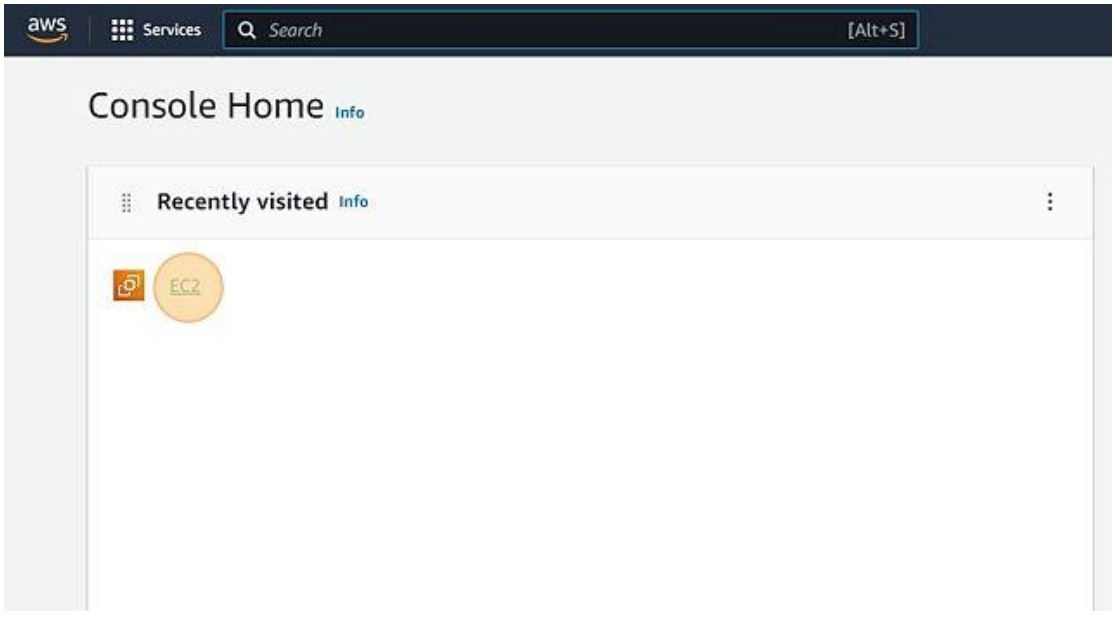
4. Click the "username@example.com" field.3



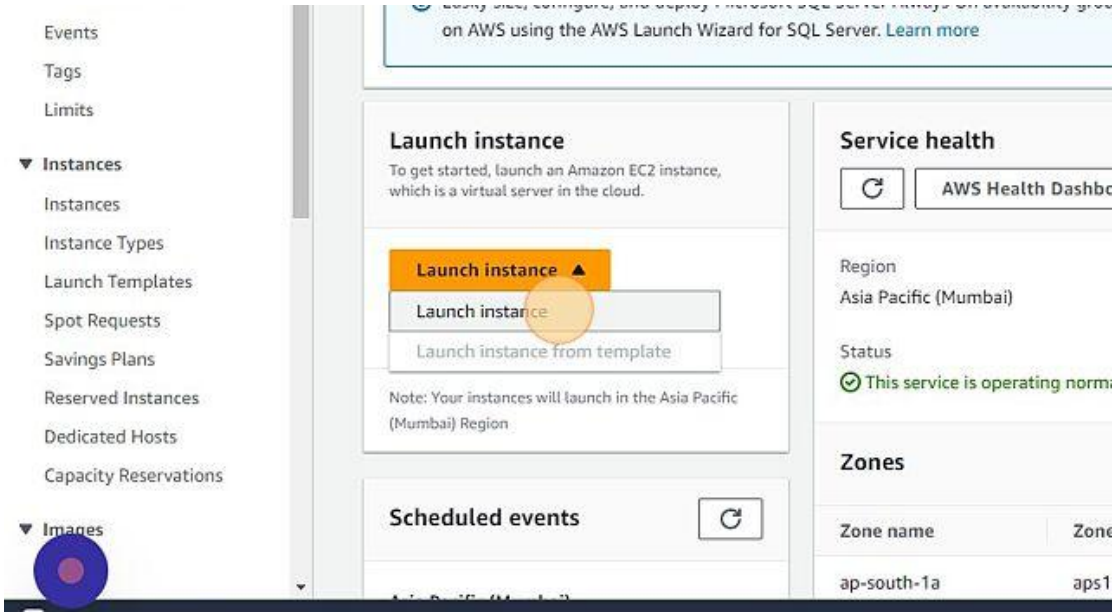
5. Click "Next"

6. Click "Sign in"4

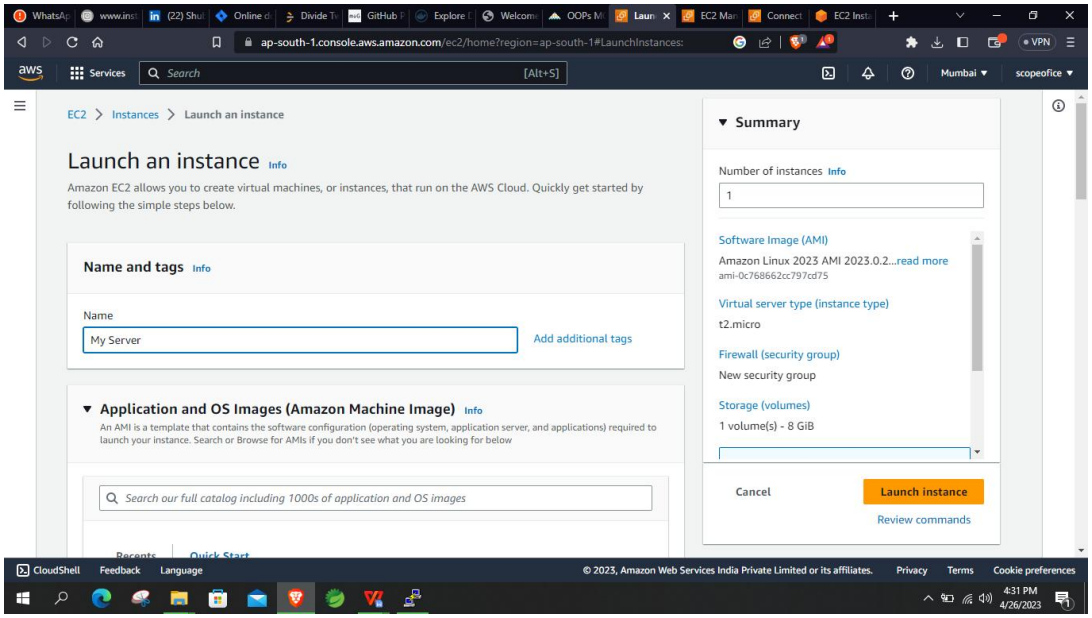
7. Click "EC2"



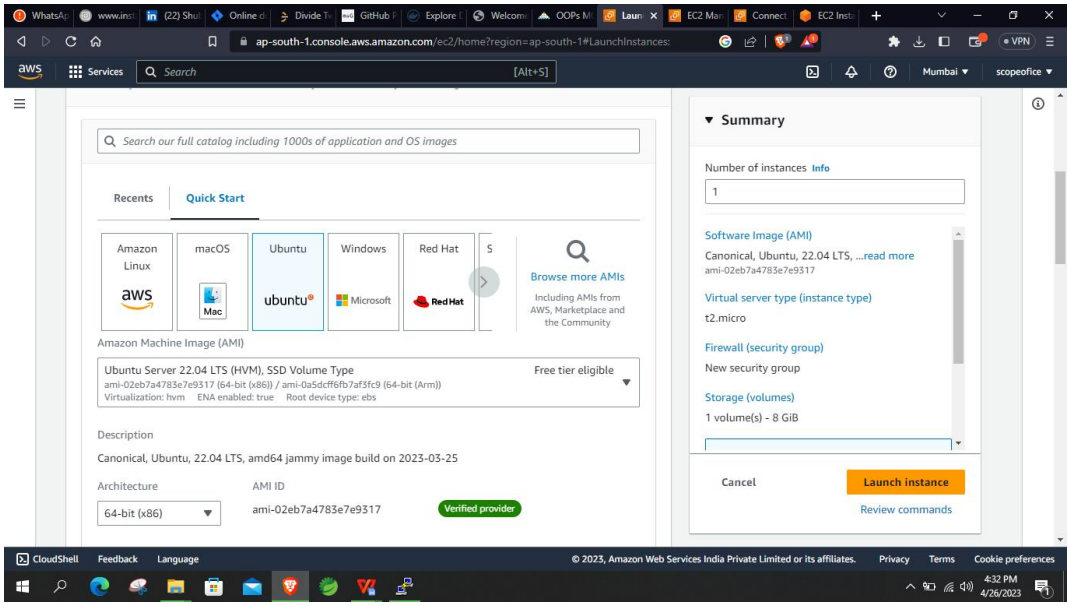
8. Click here



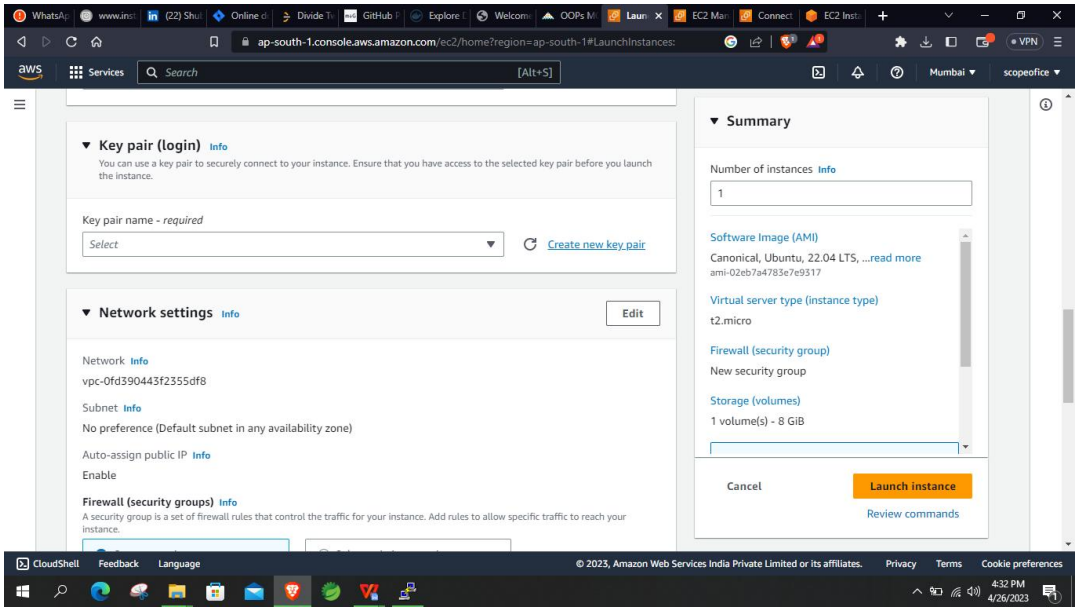
9. Give a Name



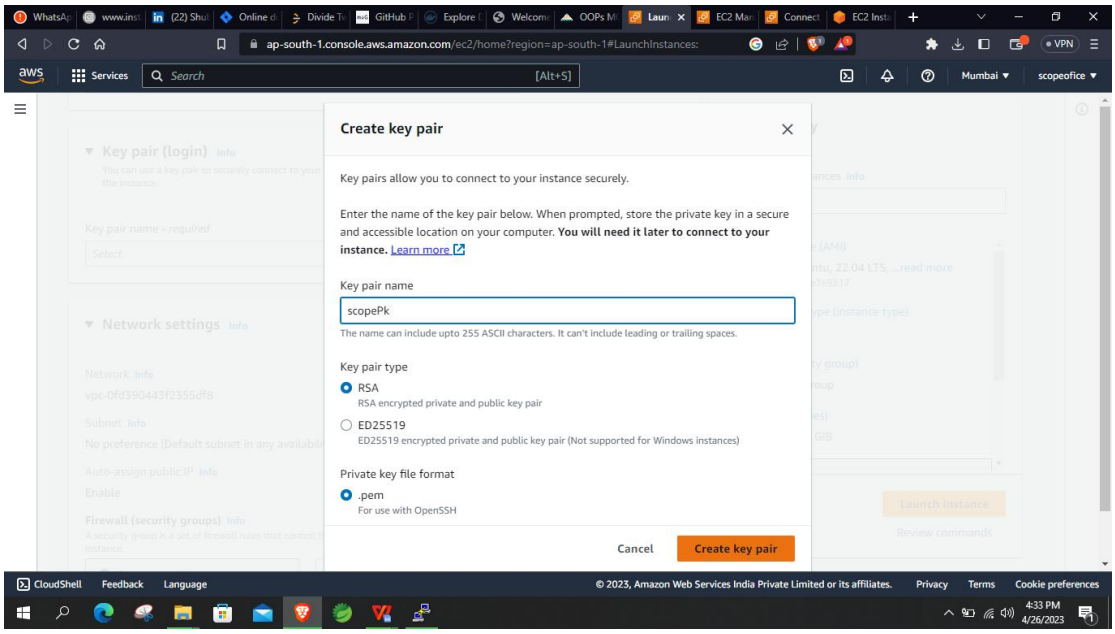
9.1 Select “Ubuntu”



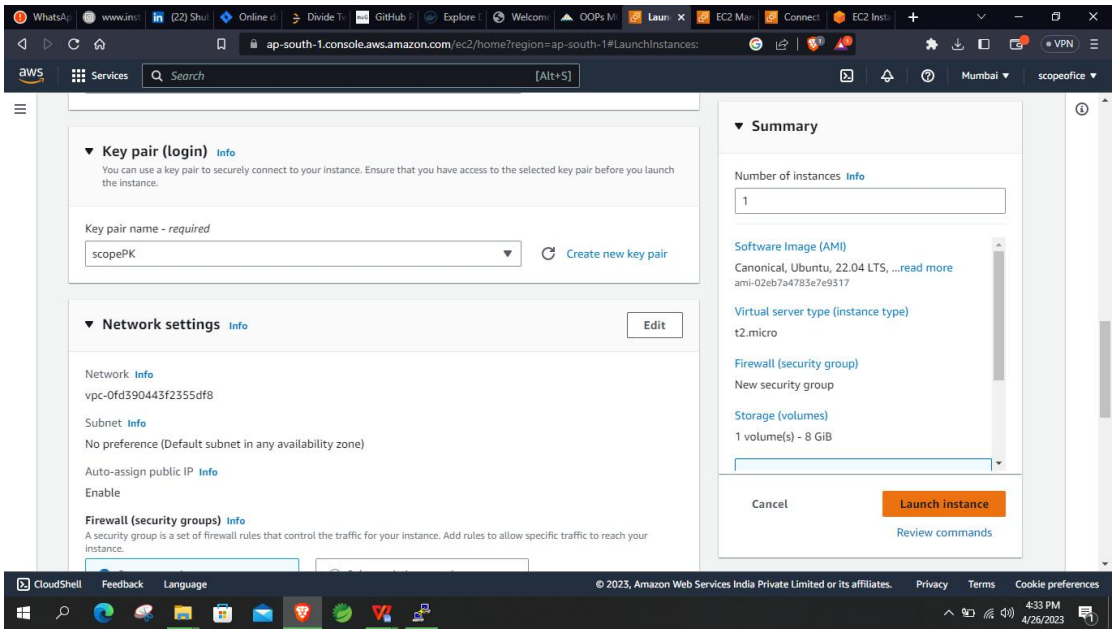
9.2 Select “Create new key pair”



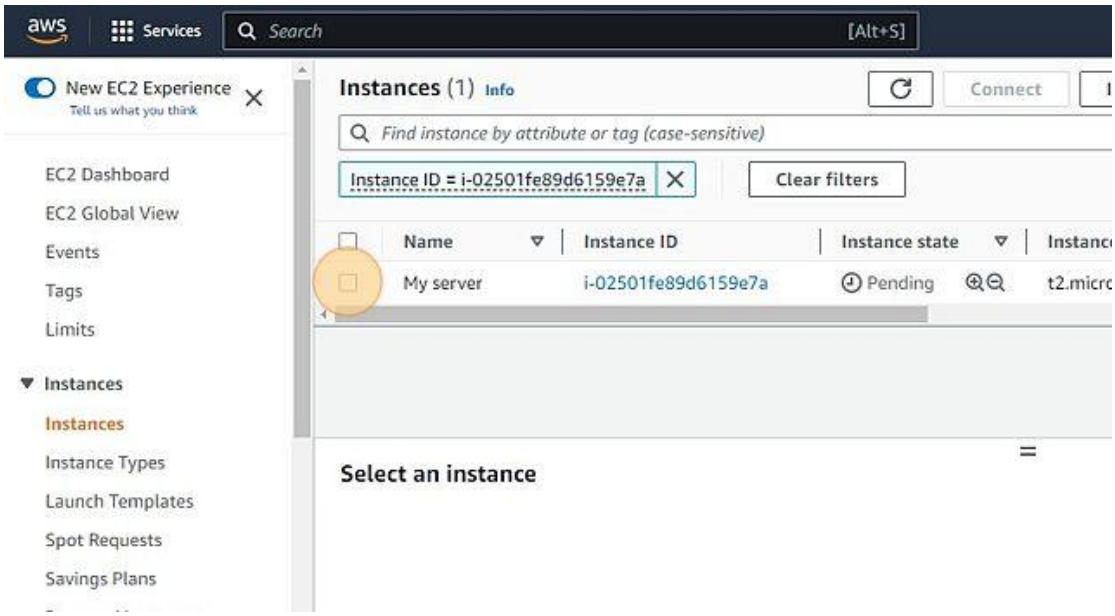
9.3 Give a name and Select “Create key pair”



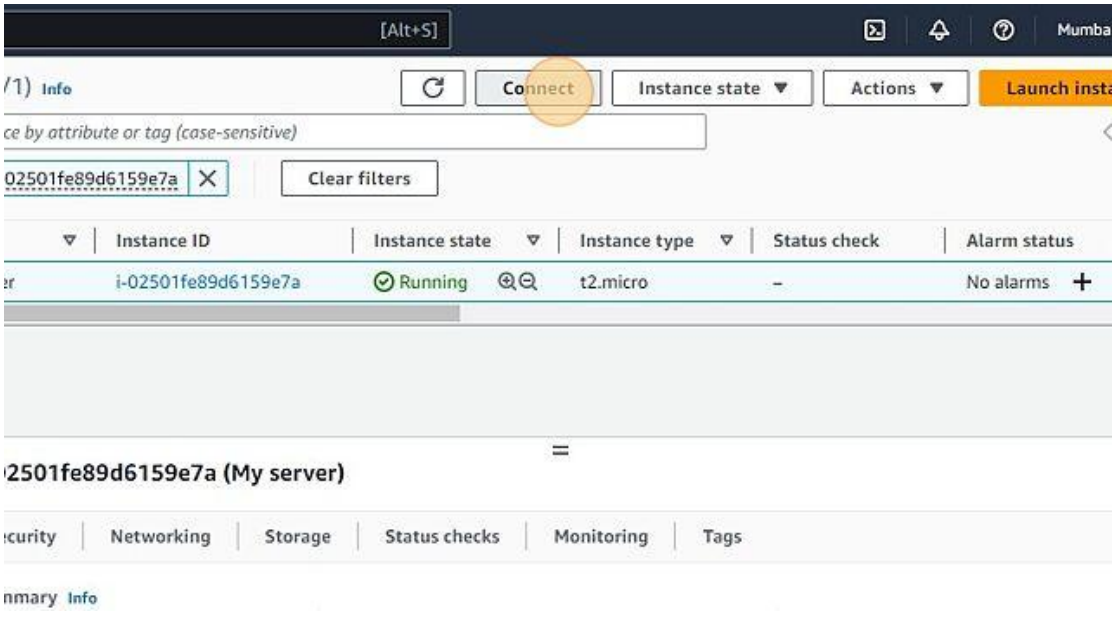
9.3 Click “Launch instance”



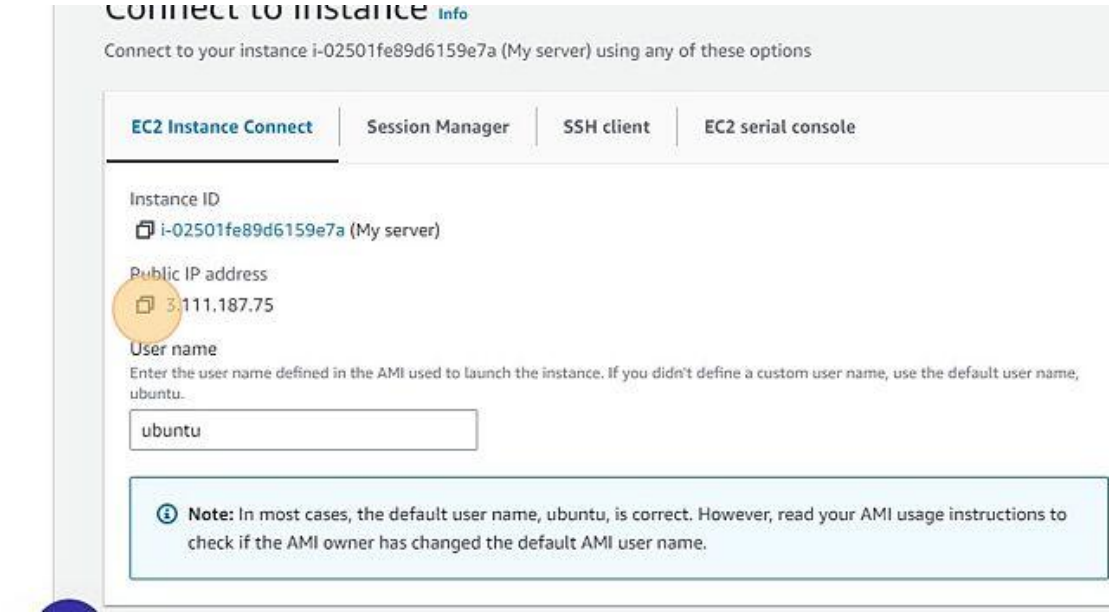
10. Click this checkbox.



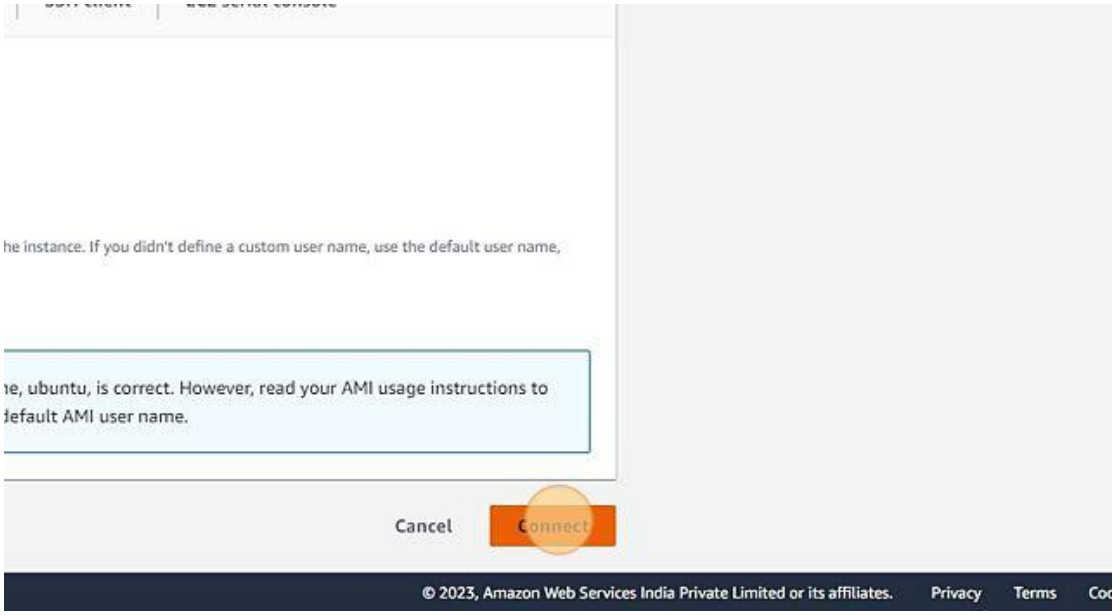
11. Click "Connect"



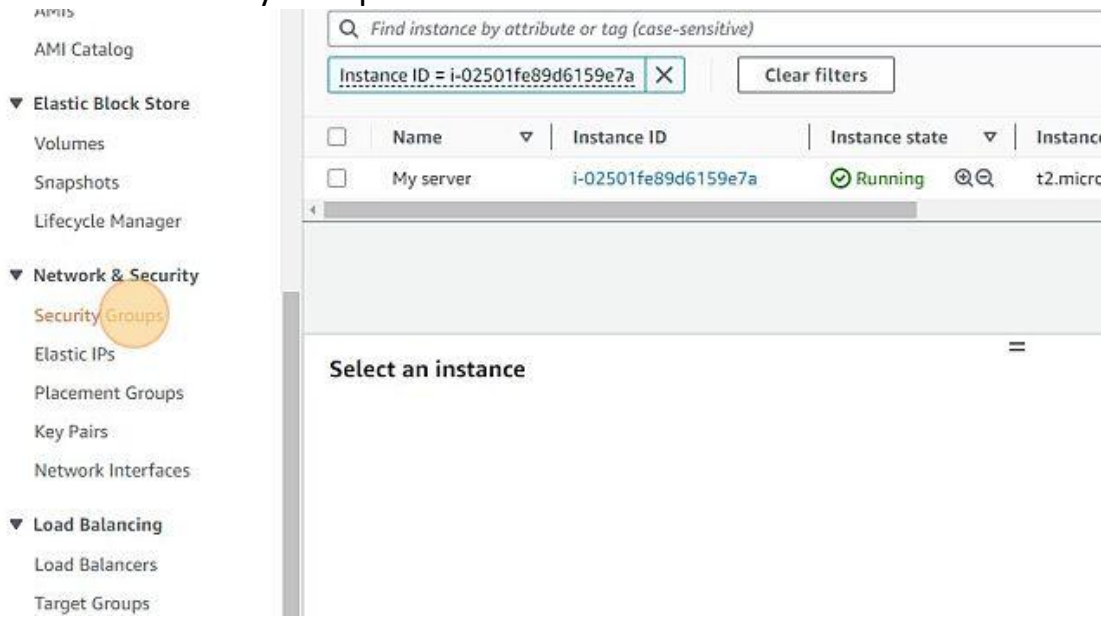
12. Click here.



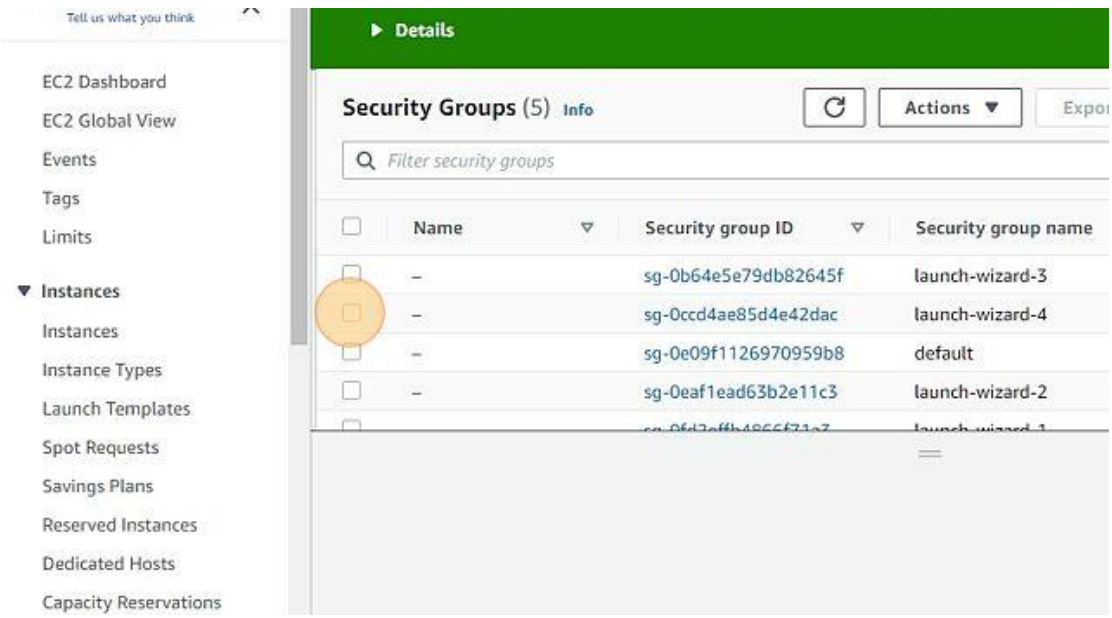
13. Click "Connect"



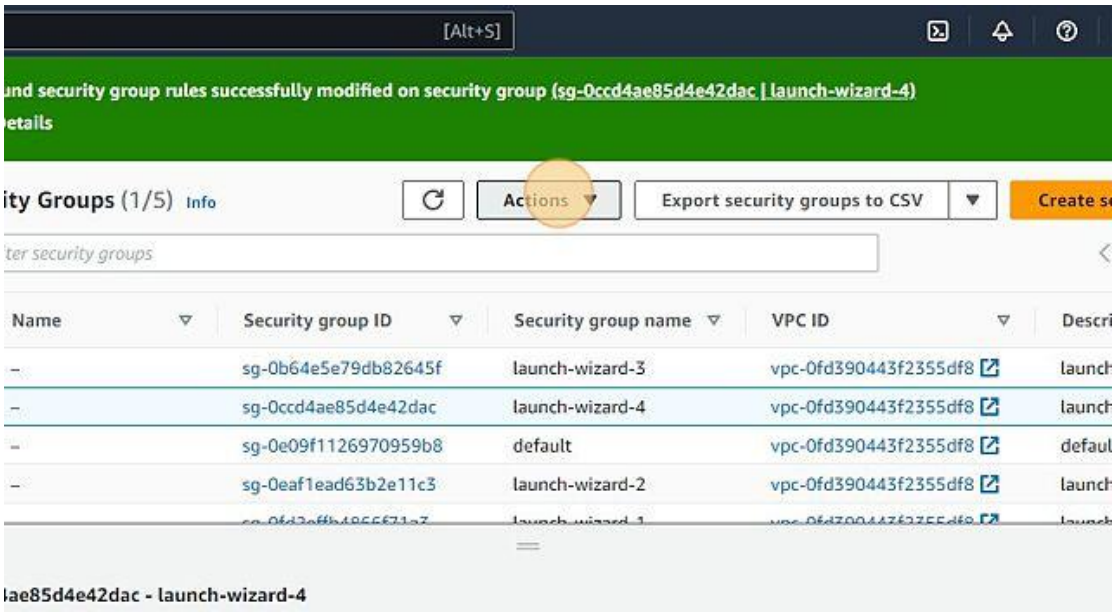
14. Click "Security Groups"



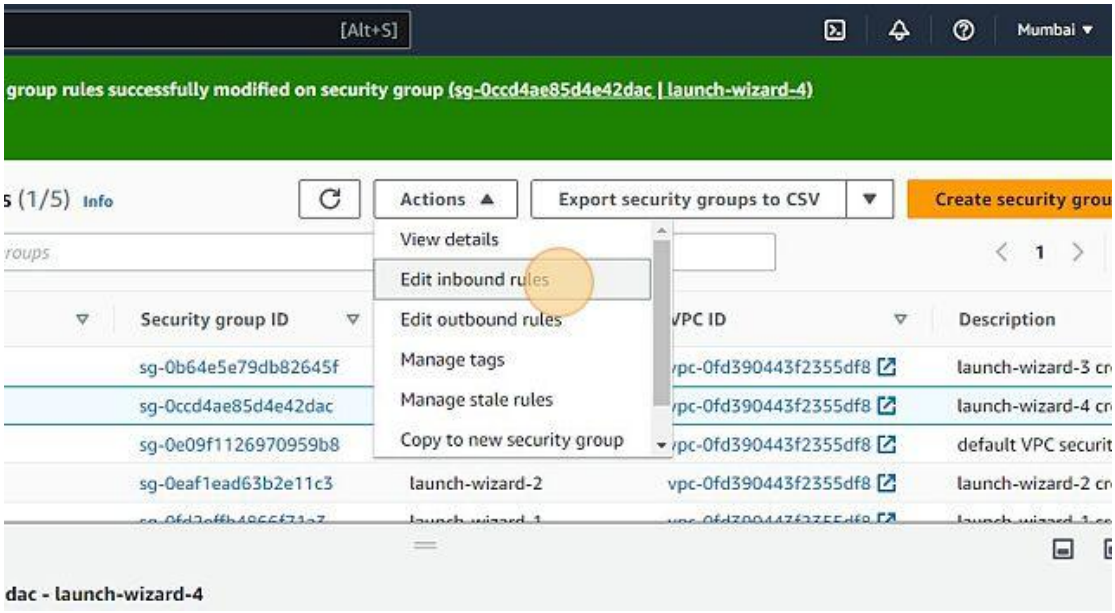
15. Select the last launch-wizard



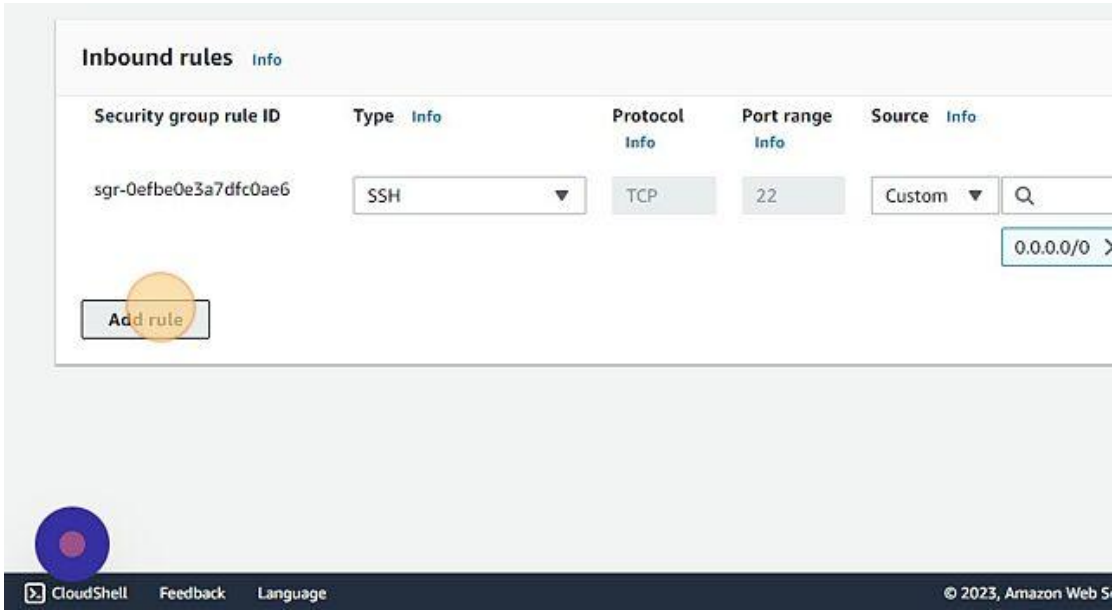
16. Click “Action”



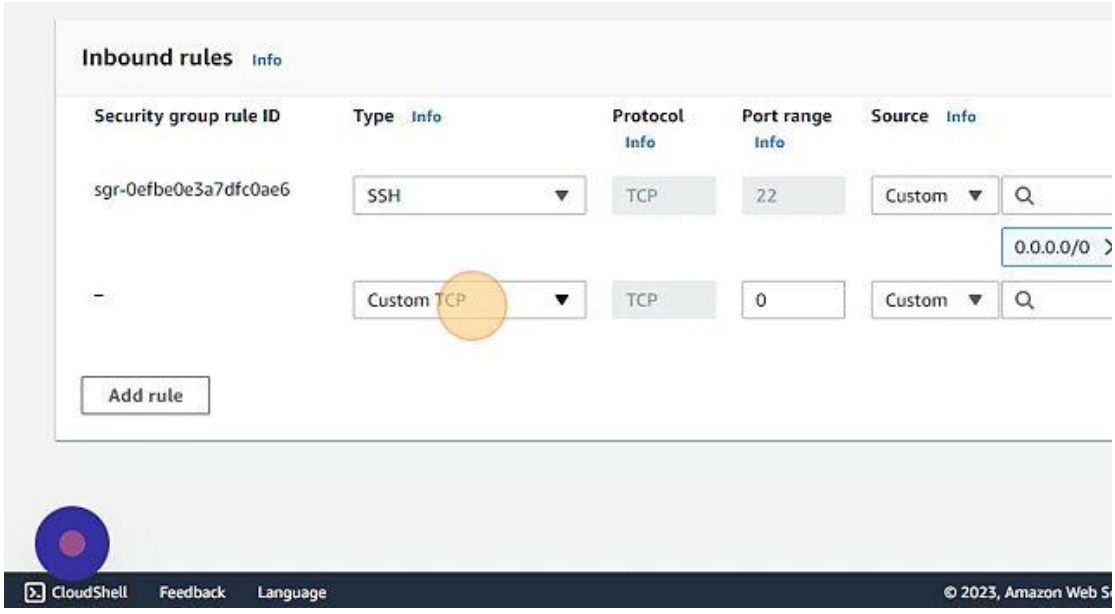
17. Click “Edit inbound rules”



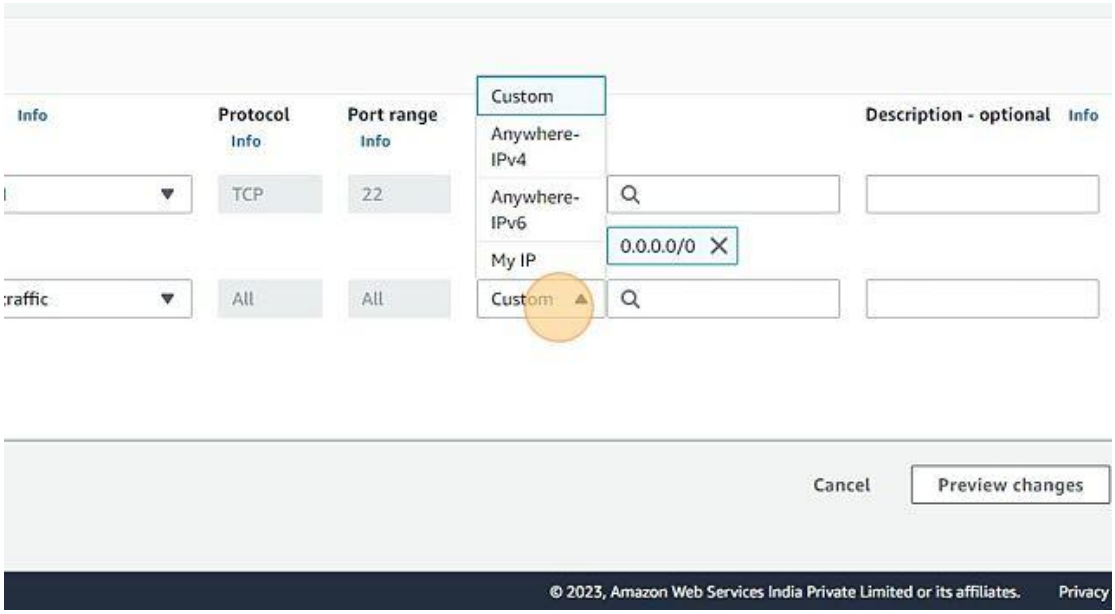
18. Click “Add rules”



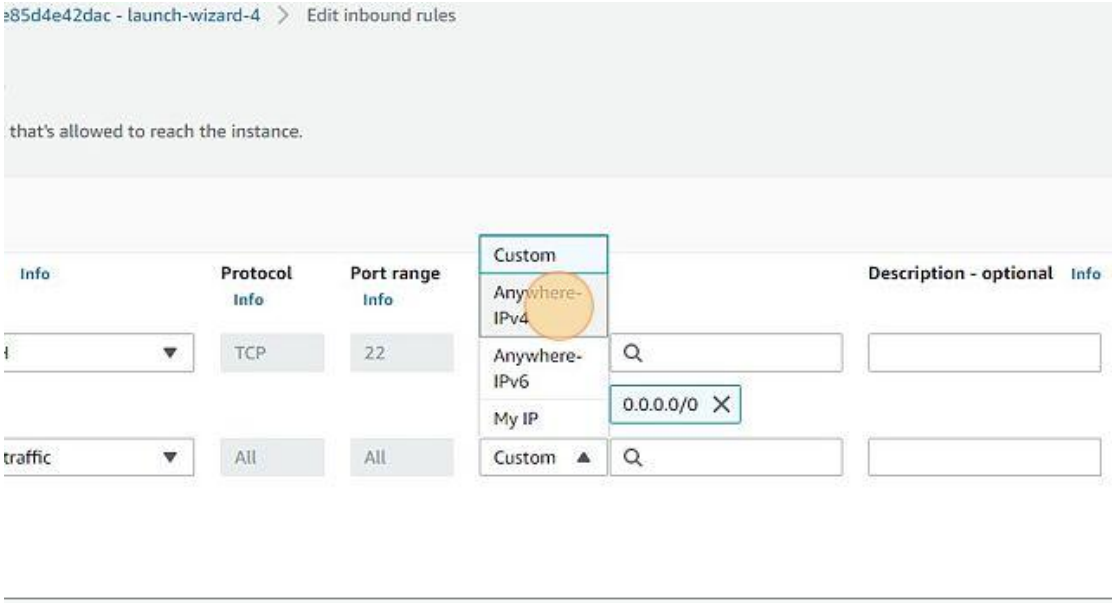
19. Click “Custom TCP”



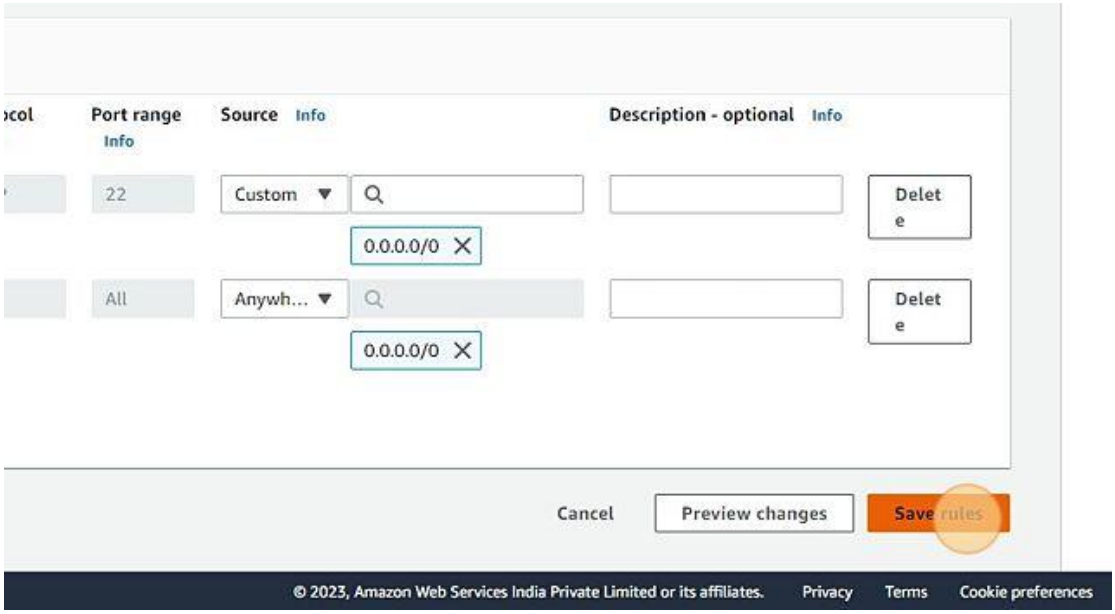
20. Click “Custom”



21. Click “Anywhere IP”

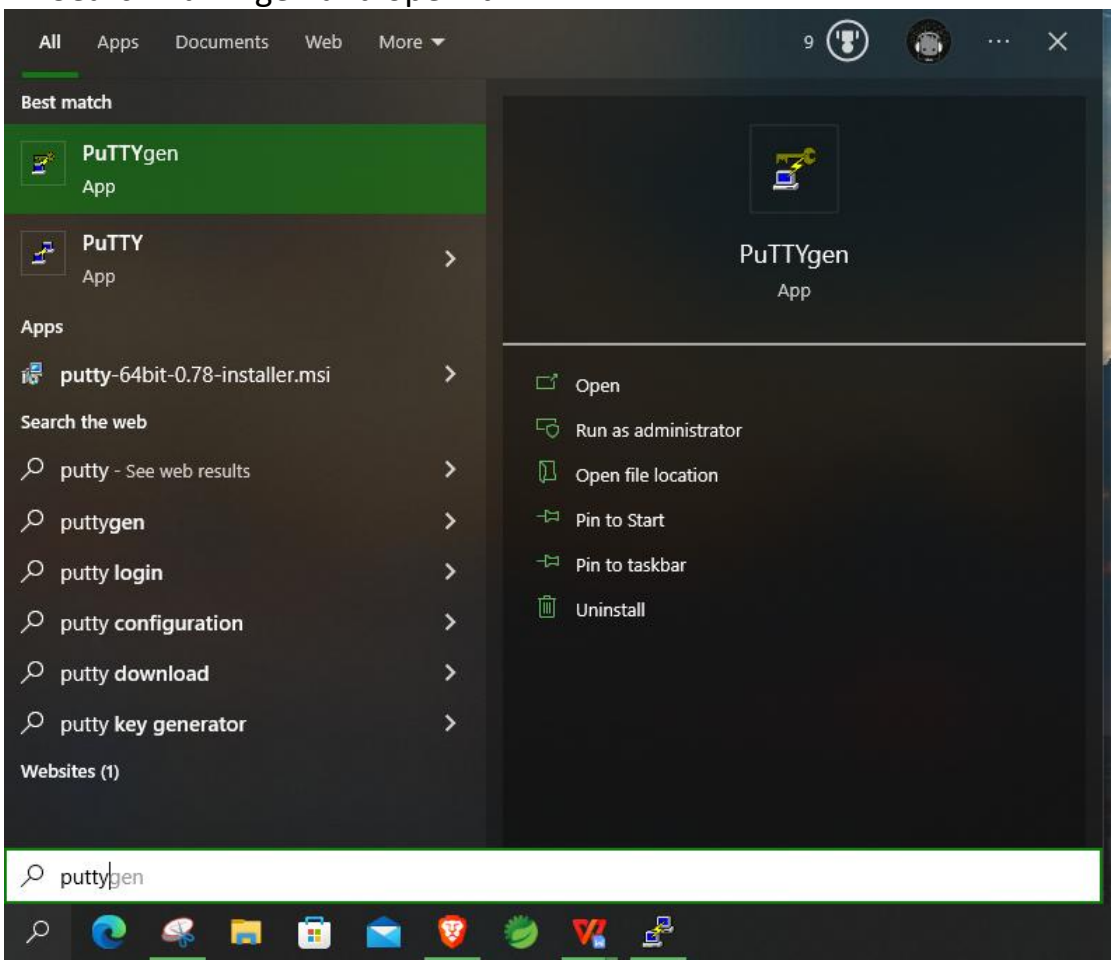


22. Click “Save rule”

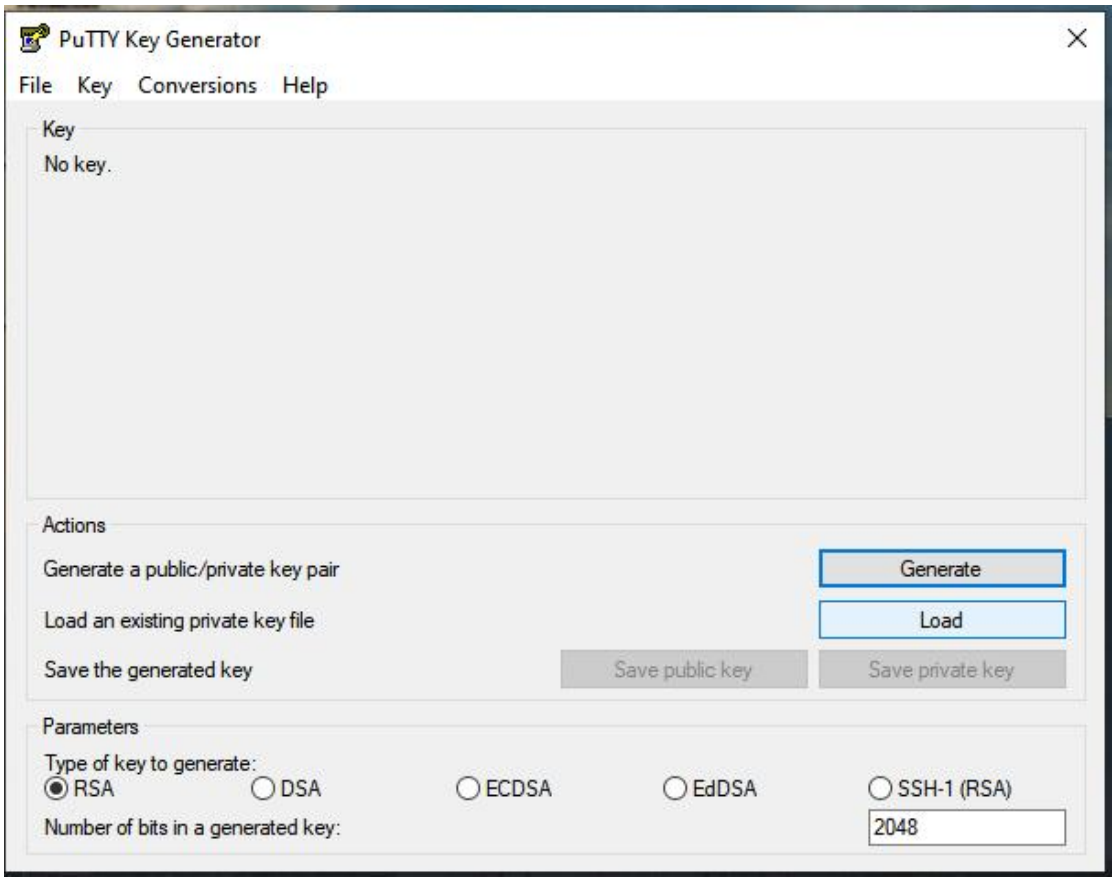


PuTTY Commands

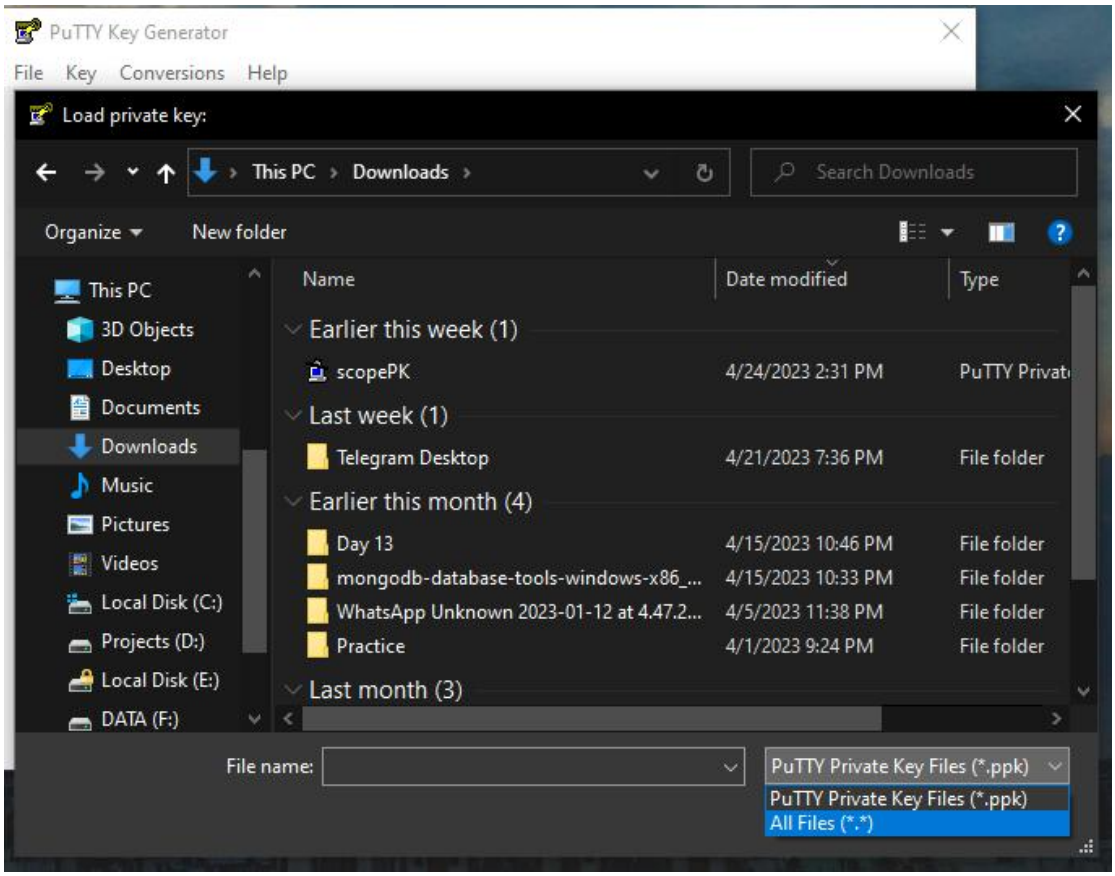
1. Search PuTTYgen and open it



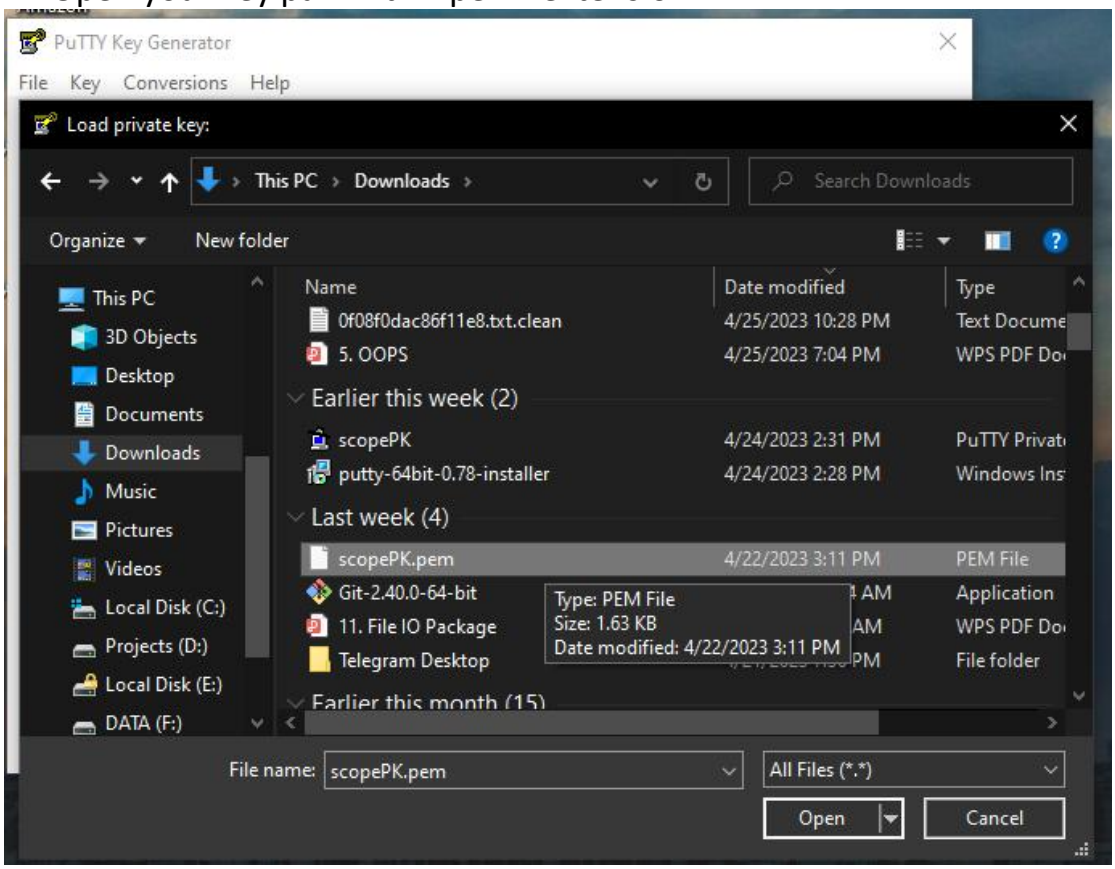
2. Click on “Load”



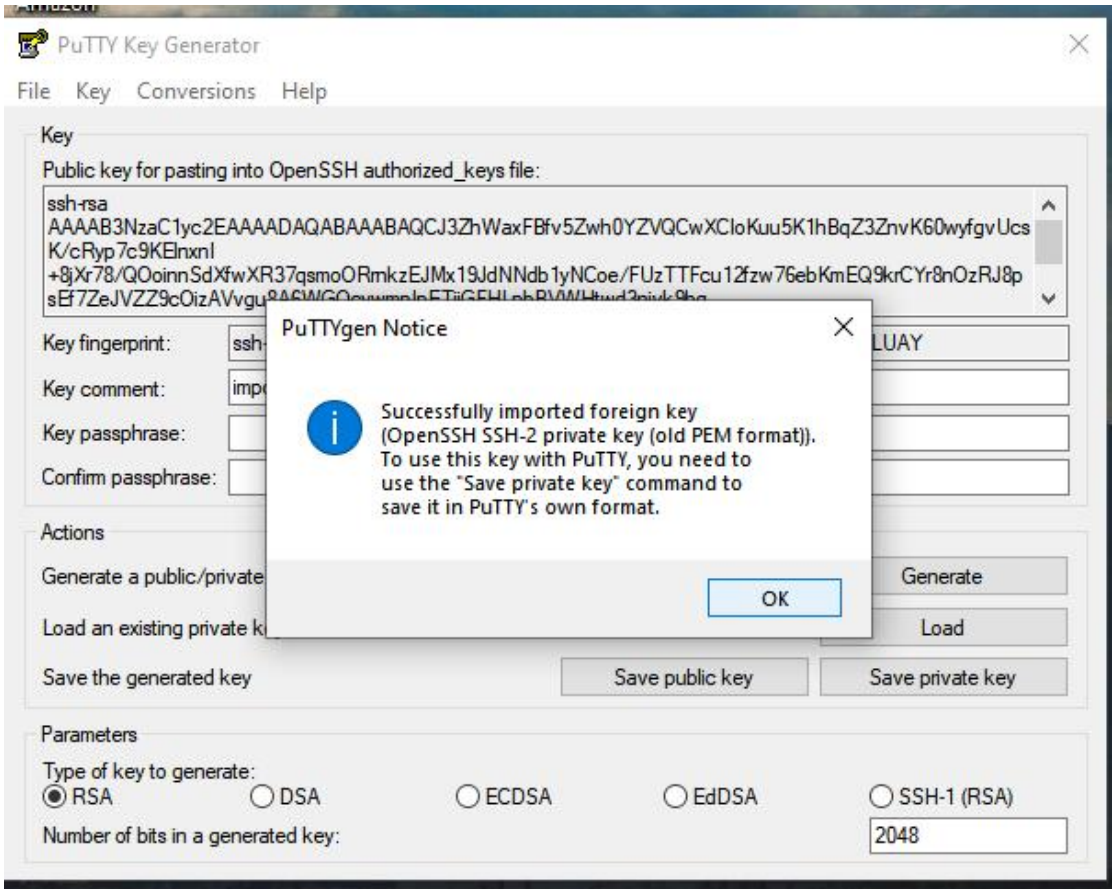
3. Select “All files*”



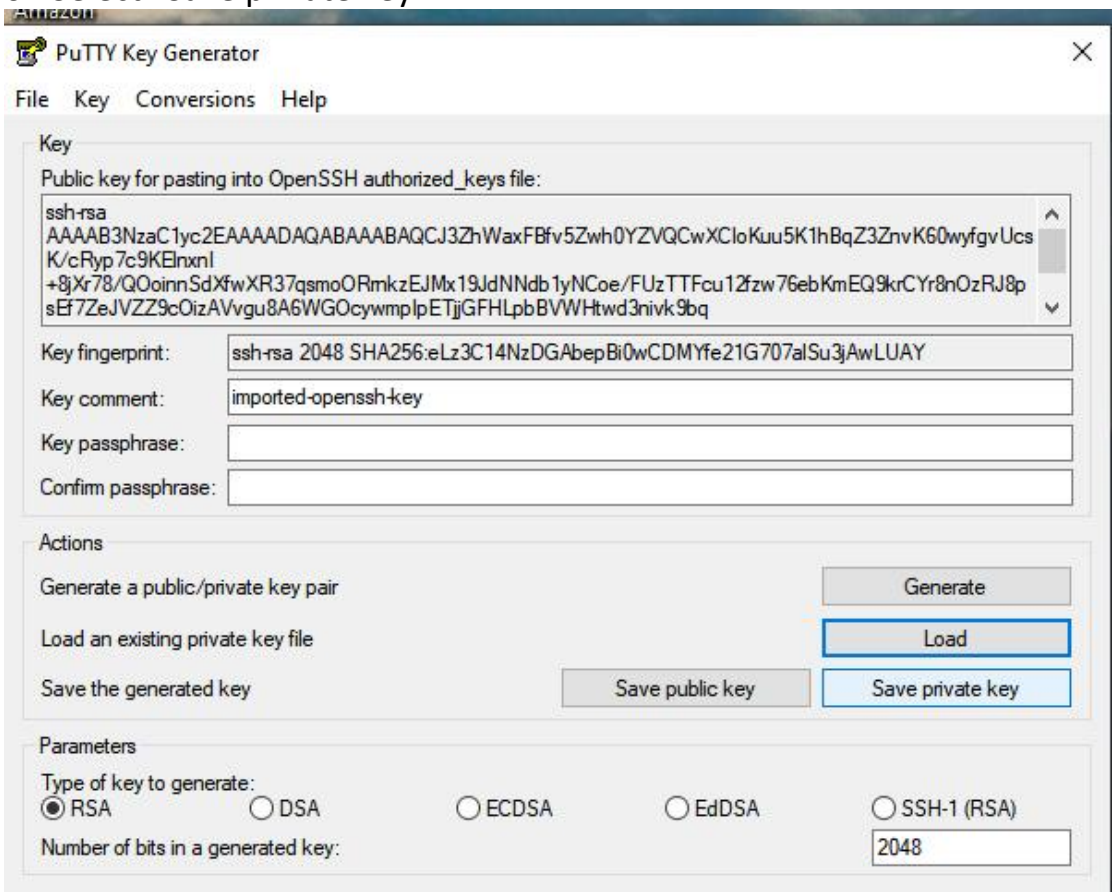
4. Open your key pair with ".pem" extension



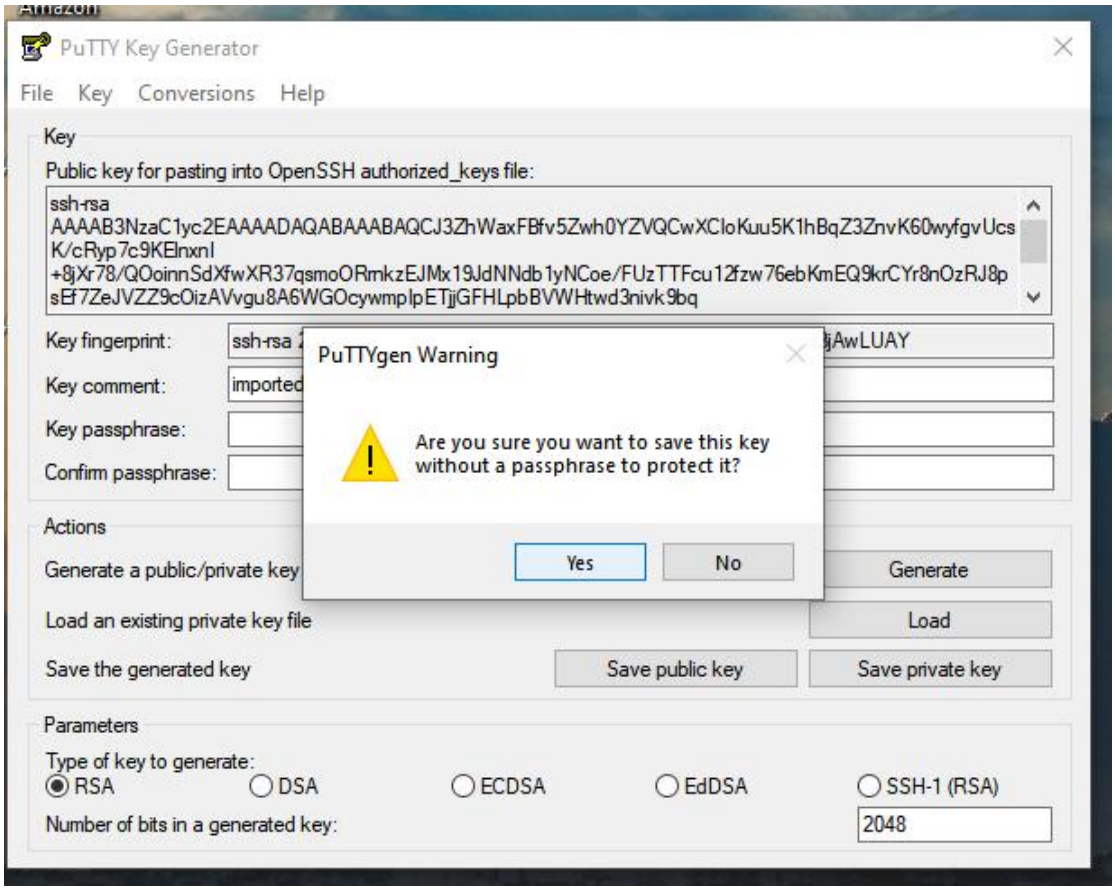
5. Click “ok”


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCJ3ZhWaxFBfv5Zwh0YZVQCwXCloKuu5K1hBqZ3ZnvK60wyfgvUcs
K/cRyp7c9KElnxnl
+8jXr78/QOoinnSdXfwXR37qsmoORmkzEJMx19JdNNdb1yNCoe/FUzTTFcu12fzw76ebKmEQ9krCYr8nOzRJ8p
sEf7ZeJVZZ9cOizAVvgu
Key fingerprint:
Key comment:
Key passphrase:
Confirm passphrase:
PuTTYgen Notice
Successfully imported foreign key
(OpenSSH SSH-2 private key (old PEM format)).
To use this key with PuTTY, you need to
use the "Save private key" command to
save it in PuTTY's own format.
OK
Actions
Generate a public/private
Generate
Load an existing private k
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
EdDSA
SSH-1 (RSA)
Number of bits in a generated key:
2048

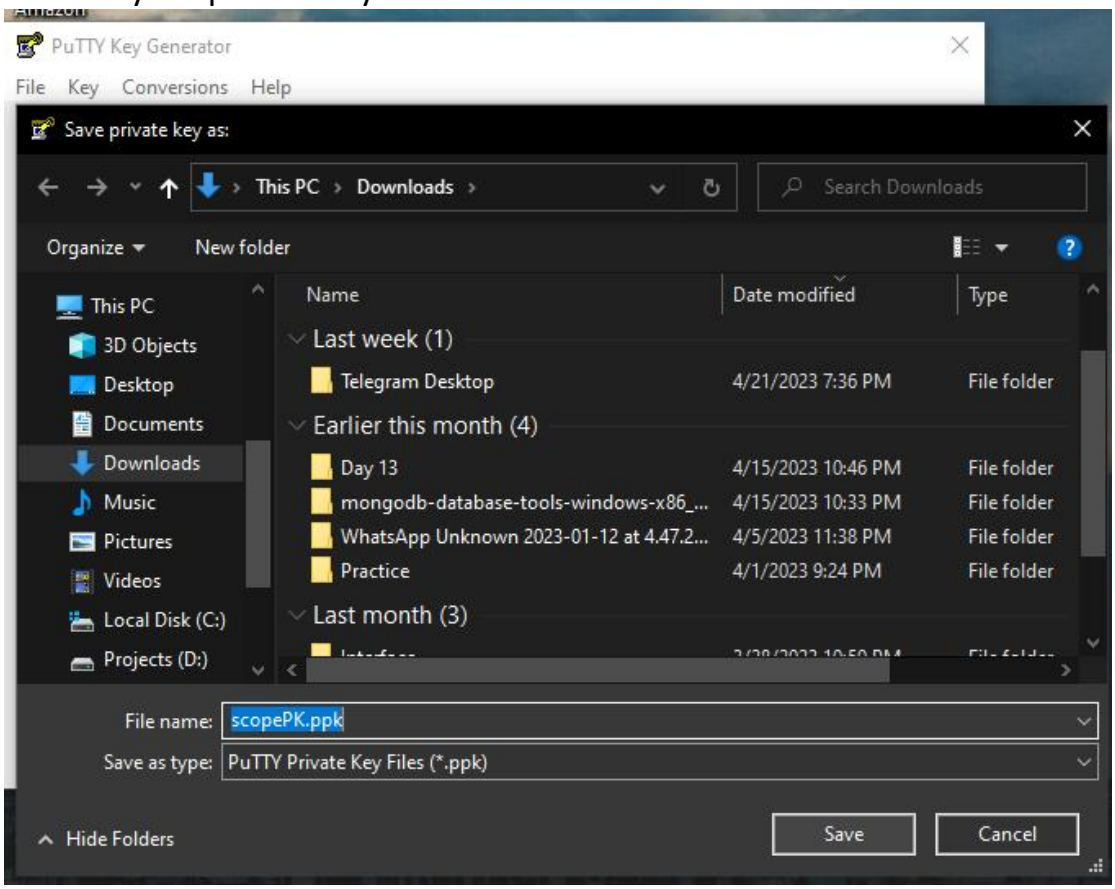
6. Select “Save private key”



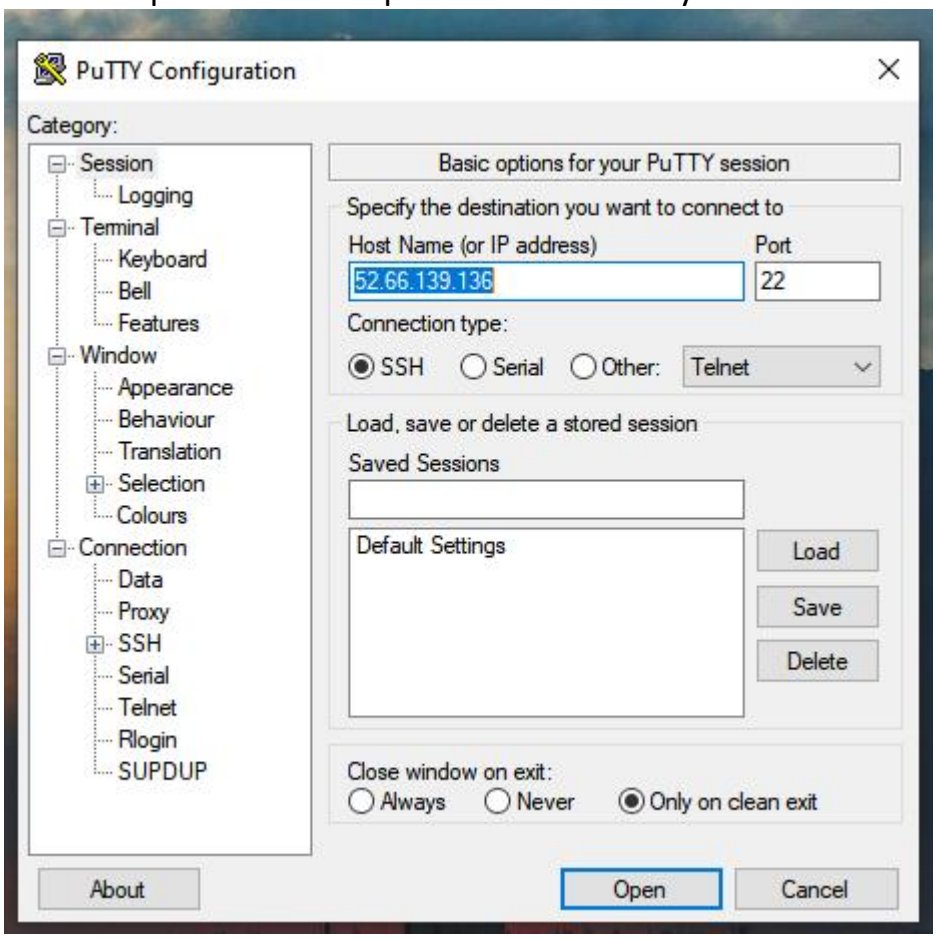
7. Click “Yes”


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCJ3ZhWaxFBfv5Zwh0YZVQCwXCloKuu5K1hBqZ3ZnvK60wyfgvUcs
K/cRyp7c9KElnxnl
+8jXr78/QOoinnSdXfwXR37qsmoORmkzEJMx19JdNNdb1yNCoe/FUzTTFcu12fzw76ebKmEQ9krCYr8nOzRJ8p
sEf7ZeJVZZ9cOizAVvgu8A6WGOcywmplpETjjGFHLpbBVWHtwd3nivk9bq
Key fingerprint:
Key comment:
Key passphrase:
Confirm passphrase:
PuTTYgen Warning
Are you sure you want to save this key without a passphrase to protect it?
Yes
No
Actions
Generate a public/private key
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
EdDSA
SSH-1 (RSA)
Number of bits in a generated key:
2048

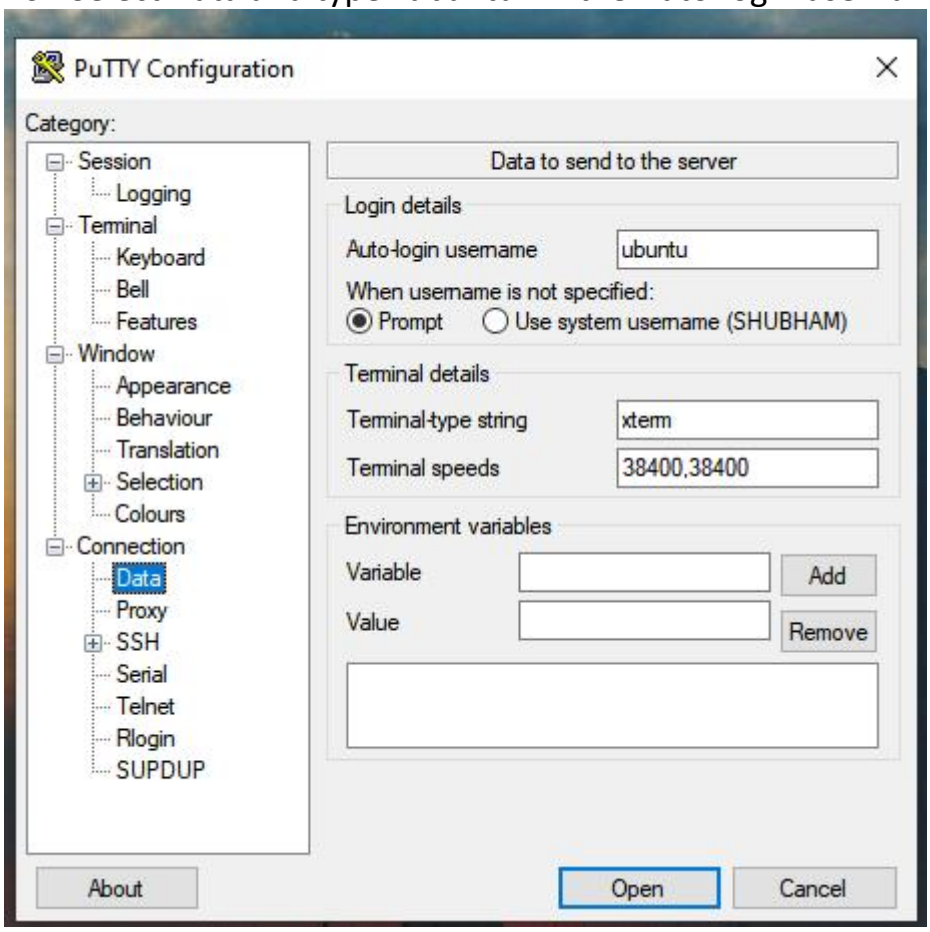
8. Save your private key



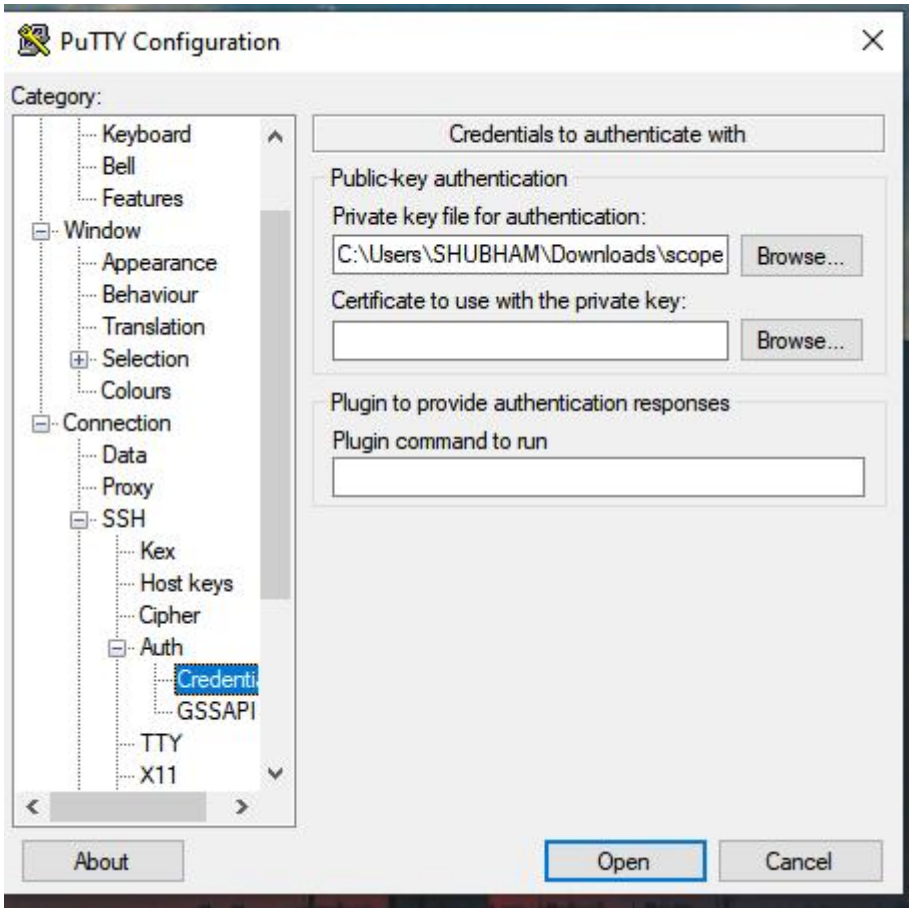
9. Now open PuTTY and paste IP address of your instance

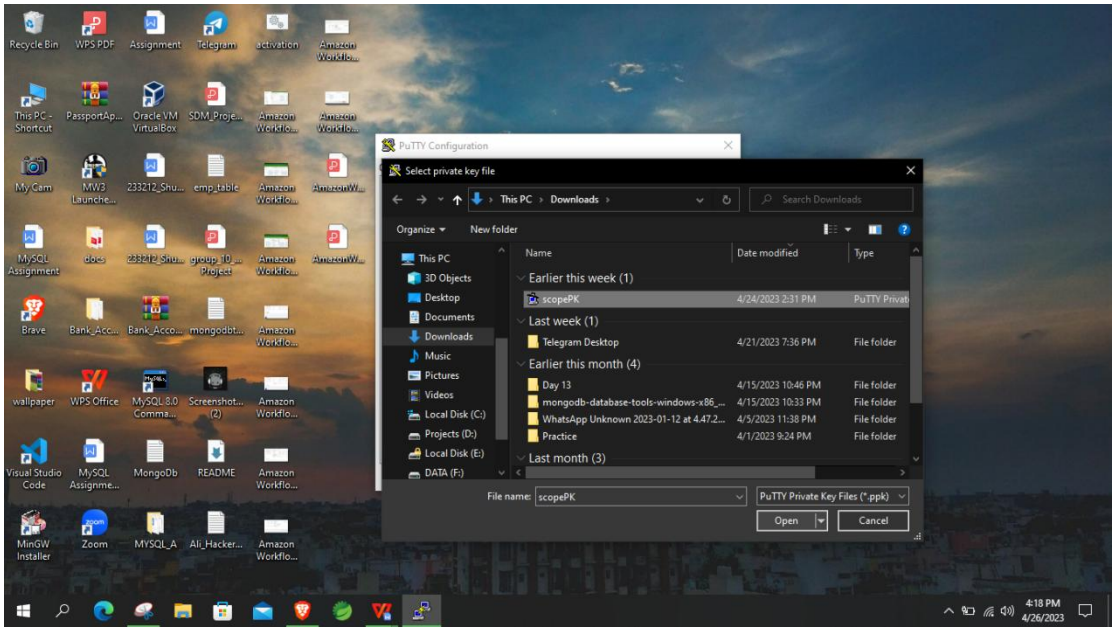


10. Select Data and type “ubuntu” in the Auto-login username

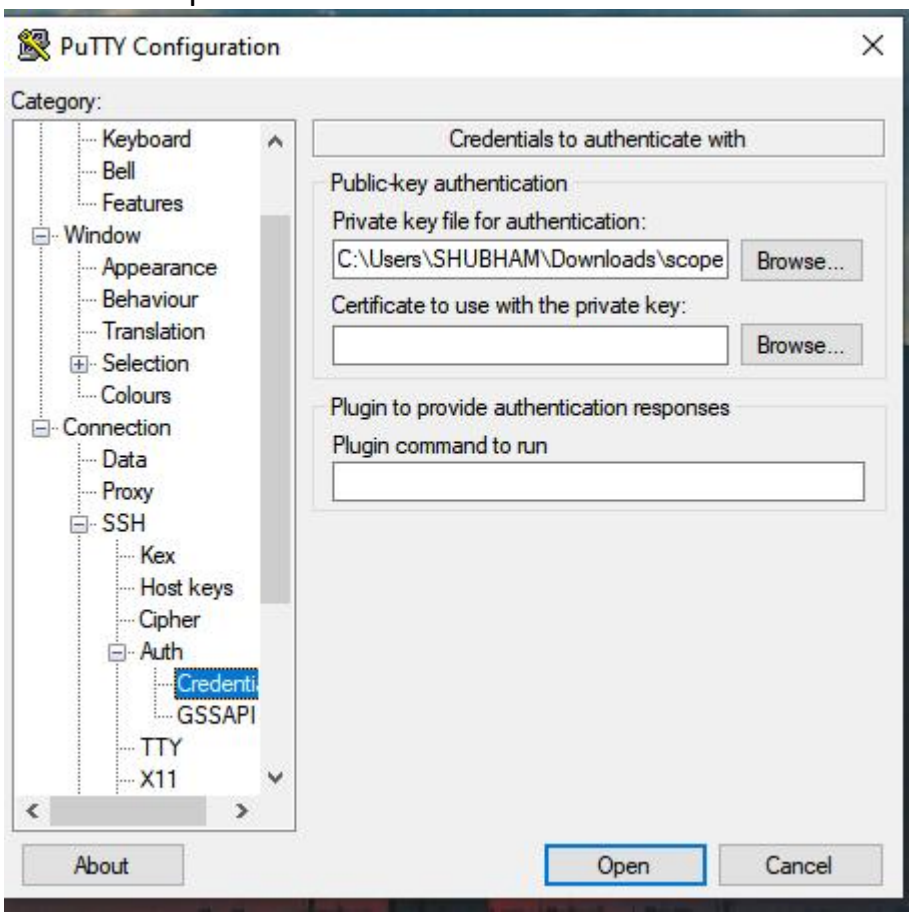


11. Go through “SSH>>Auth>>Credentials“ then browse your private key.

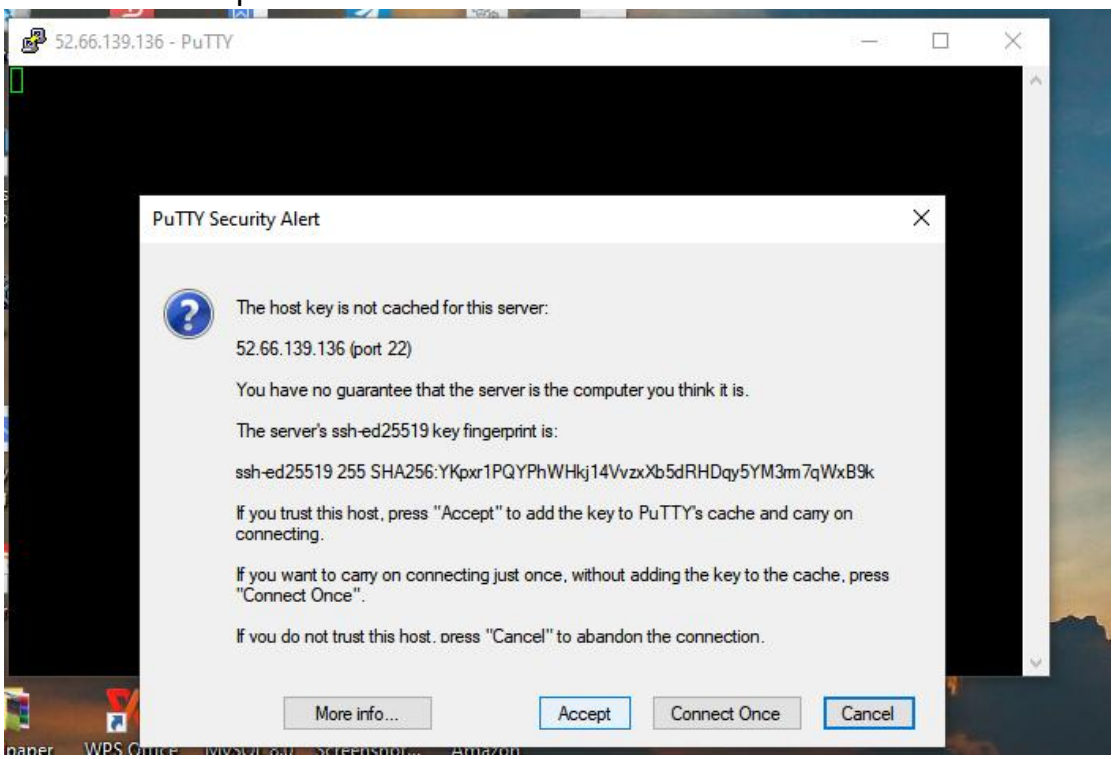




12. Click “Open”



13. Click “Accept”



14. Type command- “sudo apt install”

```
ubuntu@ip-172-31-40-213:~$ sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-40-213:~$
```

15. Type command- “sudo apt-get update”

```
ubuntu@ip-172-31-43-150:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [795 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [155 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [9024 B]
```

16. If you are using an old key, Type command- “sudo apt-get install docker.io” then “Y” else If installing for the first time- “ sudo apt install docker.io “.

```
ubuntu@ip-172-31-43-150:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse |
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 35 not upgraded.
Need to get 72.4 MB of archives.
After this operation, 287 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64
```

17. Type command- git clone https://github.com/RaviTambade/iacsdrepomarch23.git

```
ubuntu@ip-172-31-40-213:~$ git clone https://github.com/RaviTambade/iacsdrepomarch23.git
Cloning into 'iacsdrepomarch23'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 17 (delta 0), reused 14 (delta 0), pack-reused 0
Receiving objects: 100% (17/17), 76.54 KiB | 7.65 MiB/s, done.
```

18. Type command- “ Ls >> cd iascdrepomarch23 >> Ls >> sudo docker build -t iacsd . “ this is to build an image of the repo. Follow the steps carefully, you can type any name instead of iacsd and not there is a space and a dot after the name.

```
ubuntu@ip-172-31-43-150:~$ ls
iacsdrepomarch23
ubuntu@ip-172-31-43-150:~$ cd iacsdrepomarch23/
ubuntu@ip-172-31-43-150:~/iacsdrepomarch23$ sudo docker build -t iacsd .
Sending build context to Docker daemon    492kB
Step 1/7 : FROM node:7
7: Pulling from library/node
ad74af05f5a2: Pull complete
2b032b8bbe8b: Pull complete
a9a5b35f6ead: Pull complete
3245b5a1c52c: Pull complete
afa075743392: Pull complete
9fb9f21641cd: Pull complete
3f40ad2666bc: Pull complete
49c0ed396b49: Pull complete
Digest: sha256:af5c2c6ac8bc3fa372ac031ef60c45a285eeba7bce9ee9ed66dad3a01e29ab8d
Status: Downloaded newer image for node:7
 ---> d9aed20b68a4
```

19. Type command- “sudo docker images ” to check if your image is created successfully.

```
ubuntu@ip-172-31-43-150:~/iacsdrepomarch23$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
iacsd         latest    74afff4a0ef2   4 minutes ago    668MB
node          7         d9aed20b68a4   5 years ago      660MB
```

20. Type command- " sudo docker run -d -p 8081:8081 iacsd " you can give any 4 digit port number and at the end type the image name.
To check your image is running type command- " sudo docker ps "

```
ubuntu@ip-172-31-43-150:~/iacsdrepomarch23$ sudo docker run -d -p 8081:8081 iacsd
861bfbdee1262638536647b46c554779f9615e4482099e236a034ab45616e04e
ubuntu@ip-172-31-43-150:~/iacsdrepomarch23$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS
861bfbdee126   iacsd     "/bin/sh -c 'node se…"   7 seconds ago   Up 6 seconds   0.0.0.0:8081-
ubuntu@ip-172-31-43-150:~/iacsdrepomarch23$
```

21. Copy your IP address and " :8081 " type the image port no. Press enter.